AMENDMENTS TO THE CLAIMS

The following is a complete, marked-up listing of revised claims with a status identifier in parenthesis, underlined text indicating insertions, and strike through and/or double-bracketed text indicating deletions. The following listing of claims will replace all prior versions and listings of claims in the application.

LISTING OF CLAIMS

1. (Currently Amended) A method for coding a structured document, the method comprising:

generating a plurality of codes using at least one name space and allocating the plurality of codes for data types defined by the at least one name space~~name spaces~~;

assigning the at least one name space to another name space, such that at least one assignment information is generated, the assignment information including at least one inheritance relationship between the bequeathing namespace and the inheriting namespace,

wherein the assignment information of the inheriting namespace includes at least one code of a basic data type of the bequeathing namespace for a header type of the inheriting namespace,

the basic data type of the bequeathing namespace being a data type from which the header type of the inheriting namespace originates directly or the basic data type of the bequeathing name space being a basic data type of a yet another bequeathing name space in an inheritance hierarchy, the inheritance hierarchy including the yet another bequeathing name space, the bequeathing name space and the inheriting name space.

~~carrying out, for each name space, an assignment to further name spaces such that at least one assignment information is generated such that at least one~~

~~inheritance relationship is described between an inheriting name space and bequeathing name spaces; and~~

~~forming the assignment information of the inheriting name space from a list of codes of the basic types of header types of the inheriting name space, with basic types being types from which the header type originates directly or from which a header type originates, which in turn is the basic type of a header type of the inheriting name space.~~

2. (Currently Amended) The method as claimed in claim 1, wherein a subset comprising addressable data types of ~~a~~ the inheriting name space is determined based on an initial basic type of the bequeathing name space and on the basis of ~~an~~ the inheritance relationship between the bequeathing and inheriting name spaces~~ and the inheritance relationships in a name space of the basic type and the inheritance relationships in the name space of the subset~~.

3. (Currently Amended) The method as claimed in claim 2, wherein the addressable subset is determined based on ~~an~~ the initial basic type by establishing the basic types of the bequeathing name space.

4. (Currently Amended) The method as claimed in claim 3, wherein, based on the initial basic type of the bequeathing name space~~for determining the subset~~, the header type is ~~types are~~ determined in the inheriting name space~~, for which basic types are~~ and the basic type of the inheriting name space is identified from the bequeathing name space ~~by use of~~ using the assignment information, the initial basic type of the inheriting name space being a basic type ~~of the basic types~~ of the bequeathing name space.

5. (Currently Amended) The method as claimed in claim 1, wherein the assignment information assigned to the inheriting name spaces is stored together with the respective name space in a first device carrying out at least one of the coding and a decoding operation.

6. (Currently Amended) The method as claimed in claim 5, wherein the assignment information assigned to the inheriting name spaces is generated in a second device and transmitted together with the respective name space~~, in a~~ to the first device carrying out at least one of the coding and decoding operation.

7. (Currently Amended) The method as claimed in claim 1, wherein respectively separate codes~~, which are~~ independent of ~~at least one of~~ other schemas ~~and~~ or name spaces~~, for the~~ and for elements either ~~at least one of~~ defined ~~and~~ or declared in ~~at least one of~~ the schemas, name spaces, ~~and~~ or in ~~the~~ groups of ~~at least one of~~ schemas ~~and~~ or name spaces, are allocated for ~~at least one of a~~ another schema, a name space ~~and for a~~ or for another group of ~~at least one of~~ schemas and name spaces.

8. (Currently Amended) The method as claimed in claim 7, wherein, to identify the at least one of schema, name space and the group of at least one of schemas and name spaces, the respectively separate codes are sub-divided into corresponding address areas.

9. (Previously Presented) The method as claimed in claim 7, wherein the separate codes respectively comprise a local code at least one of relating to at least one of the schema and the name space and relating to at least one of the group of schemas and name spaces and an identification code to identify at least one of the

schema, name space and the group of at least one of schemas and name spaces.

10. (Previously Presented) The method as claimed in claim 7, wherein separate codes are generated for at least one of global elements, substitution groups and data types.

11. (Currently Amended) The method as claimed in claim 10, wherein separate codes are generated for data types type codes such that within the inheritance tree of a name space, the data type adjacent to a first data type in the same name space is at a code interval ~~in respect of~~ with respect to the first data type, said code interval corresponding to the number of data types derived from the first data type in this name space.

12. (Currently Amended) The method as claimed in claim 7, wherein the separate codes within a given name space are allocated according to a method comprising:

sorting all data types of a name space, which ~~were~~ are bequeathed from data types of other name spaces, in a list in ~~the~~ a sequence of global type codes of ~~the~~ respective basic data types as defined in the MPEG-7 standard, the respective basic data types being the data types in other name spaces, from which the sorted data types ~~were~~ are bequeathed;

lexicographically sorting in each instance data types of a name space, which ~~were~~ are bequeathed from a specific basic data type of a specific other name space~~,~~ ~~lexicographically in each instance~~;

sorting according to the sequence defined in the MPEG-7 standard into the existing list of data types all the data types of a name space, which ~~were~~ are not bequeathed from a data type of another name space,~~ according to the sequence defined in the MPEG-7 standard into the existing list of data types~~; and

allocating the separate codes in list sequence to the data types of the name space.

13. (Previously Presented) A method for decoding a structured document, comprising:

decoding a document previously coded according to a method as claimed in claim 1.

14. (Previously Presented) A method for decoding a structured document, comprising:

decoding a document previously coded according to a method as claimed in claim 11 wherein, to decode a binary type code, the code length of the separate codes of the binary type codes is determined from the number of derived data types.

15. (Currently Amended) A method for decoding a structured document, comprising:

decoding a document previously coded according to a method as claimed in claim 4, wherein, to decode a specific type code, ~~the~~ a sub-tree of the inheritance tree of the name space[[,]] in which the specific type code is located[[,]] is determined from the code intervals between adjacent data types.

16. (Previously Presented) A method for decoding an XML-based document, comprising:

decoding a document previously coded according to a method as claimed in claim 1.

17. (Previously Presented) A method for decoding an XML-based document, comprising:

decoding a document previously coded according to a method as claimed in claim 11, wherein, to decode a binary type code, the code length of the separate codes of the binary type codes is determined from the number of derived data types.

18. (Currently Amended) A method for decoding an XML-based document, comprising:

decoding a document previously coded according to a method as claimed in claim 11, wherein, to decode a specific type code, ~~the~~ a sub-tree of the inheritance tree of the name space[[,]] in which the specific type code is located~~, code length~~ is determined from the code intervals between adjacent data types.

19. (Currently Amended) A method for decoding an XML-based document, comprising: decoding a document previously coded according to a method as claimed in claim 11, wherein, to determine the basic types~~, which originate~~ originate from an initial basic type, code length is determined from the code intervals between adjacent data types.

20. (Previously Presented) A method for decoding an XML-based document, comprising: decoding a document previously coded according to a method as

claimed in claim 11, wherein, to determine the number of types in the subset, code length is determined based on the header types from the code intervals between adjacent header types.

21. (Previously Presented) A coding device, configured to implement a coding method as claimed in claim 1.

22. (Previously Presented) A decoding device, configured to implement a decoding method as claimed in claim 13.

23. (Previously Presented) A coding and decoding device comprising a coding device as claimed in claim 21.

24. (Cancelled)

25. (Previously Presented) A method as claimed in claim 1, for coding an XML-based document.

26. (Previously Presented) A method as claimed in claim 13, for decoding an XML-based document.

27. (Previously Presented) A coding and decoding device comprising a decoding device as claimed in claim 22.

*** END CLAIM LISTING ***